



Thank you for choosing

VRDOC Dynamic Occlusion Culling.

VRDOC is a system that hides objects that are not in the camera view frustum. This means that it effectively disables the rendering of these objects, which results in fewer draw calls and better graphics performance.

This readme is for **VRDOC v.2.0**

SETTING UP:

1. Make sure the gameobjects you want VRDOC to manage have a collider attached. The collider can be marked as trigger if you want.
2. Create a new layer named *VRDOC_ObjectLayer*.
3. Create a new layer named *VRDOC_TransparentLayer*.
4. Put all your gameobjects you want to be managed by VRDOC (including transparent ones) on the layer *VRDOC_ObjectLayer*.
5. Add **VRDOC_Camera.cs** script to your main camera. If you're using VR, this should be your eye camera.

CONFIGURING THE VRDOC_CAMERA SCRIPT

6. **VRDOC Layers** should be set as the **VRDOC_ObjectLayer** layer you created
7. **Transparency Layer** - Set is as the **VRDOC_TransparentLayer** layer you created. NOTE: No gameobjects should be on this layer. When transparent objects are visible, their layer will change from *VRDOC_ObjectLayer* to this.
8. **Raycast Field of View** - Set it to around ten units larger than your main camera field of view. A value of 120 is good for the HTC Vive and Oculus Rift.
9. **Raycast Every Nth Frame** - How many times the rays will be fired in relation to the current frame rate. A value of 1 fires the rays every frame, but will be more CPU consuming.
10. **Max Raycast Range** - How far the rays will be able to reach. Everything outside this range will be considered culled. A value of zero automatically adjusts the range to camera far clipping plane.

11. **Ray Column Count** - How many columns of rays will be inside the field of view of the main camera. VRDOC will generate a grid of rays based on this value.
12. **Object Visible Frame Time** - How many frames an object will be visible if not hit by a ray.
13. **Dynamic Sphere Size Distance Modifier (DSSDM)** - When a ray hits an object, a second ray containing a sphere will be fired along the previous ray. The size of the sphere is calculated by multiplying the distance of the ray hit point with DSSDM.
14. **Static Sphere Cast Radius** - The radius of the sphere going along a ray if DSSDM is not used.
15. **USE REALTIME SHADOWS** - Set to true if your game has realtime shadows.
16. **USE DYNAMIC SPHERE SIZE** - Set to false if you don't want to use distance based sphere sizing (DSSDM). This option is recommended to be turned off for mobile platforms.
16. **VISUALIZE CULLING** - (EDITOR ONLY) Shows debug information about culling.
17. **DISABLE LOGGING** - Disables VRDOC log messages.

ADDITIONAL NOTES

- **VRDOC_Object** script is automatically added to all gameobjects that reside on **VRDOC_Layers** (specified in VRDOC_Camera).
- The **isTransparent** boolean in the **VRDOC_Object** script specifies if an object can be seen through. If your game uses the Unity Standard shader, this flag is set automatically by VRDOC if any of the materials are marked as "Transparent". If you use a custom shader, you can add the VRDOC_Object component manually to a gameobject and set **isTransparent** to **true**.
- You should not have any gameobjects on the **VRDOC_Transparent** layer. This layer is used as a temporary container to allow rays to pass through the object.

SUPPORT

If you have any questions, don't hesitate to contact me at anton.korhonen@viversion.com.



<https://www.viversion.com/contact/>
<https://www.facebook.com/viversion>



<https://twitter.com/ViversionGames>
<https://www.instagram.com/viversiongames>



<https://www.youtube.com/channel/UCNtGbApCw1iI7e8T-30ImnQ>